IN THE SPECIFICATION:

- Please replace the following paragraph beginning at page 7, line 13 as follows:

Figure ~~2~~ 3 is a graph illustrating the results of combining two adaptation algorithms, the HMM adaptation, and the Word Bigram Statistics adaptation.

- Please replace the paragraph beginning at page 14, line 19 to page 15, line 9 with the following paragraphs and tables:

For these experiments, the ASR engine's first-pass Viterbi search graph was biased with word bigram data extracted from subsets of the recognition results on given development input data. The result transcriptions (i.e., recognition hypotheses) were randomized and then various sized portions, starting from the top-choice down, were taken to accumulate word pair frequencies. A variation on this experiment imposed a score threshold on the recognition results as the sub-setting mechanism. These counts were converted into probabilities, and these probabilities were used to bias the Viterbi search in favor of the most likely word sequences. The main results of these experiments are summarized in the ~~table~~tables 1A and 1B illustrated ~~in Figures 3A-3B.~~ below:

Table 1A: Recite First-Pass Results on AT&T Dev, Baseline (B) vs. Adapted (A/S) Word Bigram Priors (A = All Dev Test Data, S = Subset Result for confidence > 0.2)

			Word Stats					Sent Stats			
	WER	SER	Corr	Rej	Sub	Del	Ins	Rej	Sub	#Wrd	#Snt
B	34.07	38.81	72.75	1.31	22.61	3.33	6.82	1.19	37.62	10281	4447
A	28.04	31.21	77.86	1.31	17.84	2.99	5.90	1.19	30.02	10281	4447
S	30.21	33.96	75.52	1.31	19.82	3.35	5.73	1.19	32.76	10281	4447

N	Inclusion Baseline	Adapted	Subset
1	61.19	68.79	66.04
2	71.73	77.38	75.20
3	76.03	80.50	78.86
5	79.78	83.00	82.51
10	83.79	85.47	85.90
20	86.37	87.23	88.33
50	87.07	87.79	89.07

CPU Performance

	#sent /utt	#seg /utt	#bnd /utt	#pushes /utt	#usecs /utt	RT-MHz (P233)
B	4394	353.0	60.3	7.575e+05	1.883	85.1
A	4394	353.0	60.3	5.914e+05	1.889	85.4
S	4394	353.0	60.3	6.539e+05	1.983	89.6

Table 1B: Recite First-Pass Results on AT&T Eval, Baseline (B) vs. Adapted (S) Word Bigram Priors (S = Subset for confidence > 0.2)

	Word Stats WER	SER	Corr	Rej	Sub	Del	Ins	Sent Stats Rej	Sub	#Wrd	#Snt
B	34.89	39.55	72.14	0.87	23.49	3.49	7.04	0.89	38.66	10016	4402
S	31.38	35.14	74.67	0.87	20.92	3.54	6.05	0.89	34.26	10016	4402

N	Inclusion Baseline	Adapted	Subset
1	60.45	64.86	
2	72.01	75.28	
3	76.37	79.15	
5	80.33	82.80	
10	84.03	86.12	
20	86.94	89.10	
50	87.71	89.66	

CPU Performance

	#pushes /utt	#usecs /utt	RT-MHz (P233)
B	7.575e+05	1.883	85.1
S	6.443e+05	1.862	84.2

Please replace the paragraph beginning at page 17, line 13 with the following paragraph and tables:

04

The ~~tables illustrated in Figures 4A-4B~~ **tables 2A and 2B** below show the experimental results for the Names task:

TABLE 2A **Recite First-Pass Results on IVPATT613 Dev, Baseline vs. Adapted Transcriptions**

UC 1	UC 2	UC 3	UC 10	UC 20	Lexicon*
86.3	91.4	93.1	95.3	96.0	baseline
88.0	93.2	94.7	97.3	97.8	reqnames_dt+std.fa_prons.+std
88.5	93.5	95.0	97.4	97.8	gen+names_dt+std.fa_prons.+std

TABLE 2B **Recite First-Pass Results on IVPATT613 Eval, Baseline vs. Adapted Transcriptions**

UC 1	UC 2	UC 3	UC 10	UC 20	Lexicon*
89.1	93.2	94.5	96.6	97.2	baseline
89.2	93.4	94.6	97.3	97.9	reqnames_dt+std.fa_prons.+std
89.4	93.9	95.1	97.3	98.1	gen+names_dt+std.fa_prons.+std

* Lexicon Descriptions:
reqnames_dt+std.fa_prons.+std:
request_names trained DT to generate alternate transcriptions
added standard transcriptions (official IVPATT dev dict)
filtered with a forced alignment over the DEV data
added standard transcriptions again to create final recognition lexicon
gen+names_dt+std.fa_prons.+std:
same as reqnames, except used gen+names to train DT

Please replace the paragraph beginning on page 18, line 7 as follows:

As described earlier, each of the above approaches is fairly orthogonal. Thus, two or more of these adaptation methods may be combined to produce an additive benefit. A simple combination of the HMM and Bigram adaptation generated the results illustrated in ~~Figure 5~~ **Figure 3.** Combining these adaptation algorithms had a somewhat less than additive effect on the development test data, and a somewhat more than additive effect on the evaluation test data. The combined improvement in both cases approached 18%.